



Ubuntu as technical artifact, on-line community and technical infrastructure

Fieldwork Report - Andreas Lloyd - December 2006.

4270 words (ex. notes, appendixes and references)

• • • •

Table of Contents

Socio-cultural setting of the fieldwork.....	2
Aim of the fieldwork.....	4
The practical course of the fieldwork.....	5
Fieldwork methods and results.....	6
On-line methods.....	6
In-person methods.....	8
Analytical perspectives.....	11
Appendix A – Glossary.....	13
Appendix B – Fieldwork initiation announcement.....	16
Appendix C – Course of fieldwork – April to November 2006.....	18
Appendix D – Ubuntu Census Survey announcement.....	20
Appendix E – Fieldwork visit request.....	21
Appendix F – Interview Question Guide.....	22
References cited.....	25

Socio-cultural setting of the fieldwork

In the period between April and November 2006 I conducted an ethnographic fieldwork in the on-line community around the Free Software project Ubuntu.

Free Software (also known as Open Source Software¹) is software licensed under a non-restrictive license that gives any user of the software the *freedom* to access, modify, and redistribute the source code. These aspects of Free software usually result in an open mode of community development where distributed volunteer software developers with similar interests collaborate on software projects over the Internet, writing and sharing the *source code*, integrating the efforts of casual contributors and users in order to gradually and organically improve the software in a meritocratic fashion.

This mode of development has a native appeal to many computer enthusiasts as they seek to share their mastery of the computer, and they often label themselves *hackers* to reflect this. This is not to be taken in the sense often used by the press to describe computer criminals, but rather as “A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary” (cf. *Jargon File* 2006).

These Free Software projects often become interdependent through their use of software licenses such as the *GNU General Public License* which formalizes the sharing of knowledge by requiring any software modified under that license to be redistributed under the same terms as stated in that license and thus disallowing the use of GPL-licensed software within closed-source software. To some extent, this formalized reciprocity defines the whole on-line Free Software eco-system of which Ubuntu is part.

The Ubuntu project was initiated by South African IT millionaire Mark Shuttleworth in 2004 with the goal of integrating the best of the (per definition freely available) Free Software into a desktop system usable by as many people as possible as a viable, free alternative to Microsoft Windows.²

Shuttleworth initially hired 12 high-profile developers from a number of Free Software projects through his company Canonical, and based on their experience and technical knowledge, particularly

1 The two different terms define an ideological divide between the free software and open source software movements. The term Free Software preceded the term open source software, and was first used by the hacker Richard Stallman and his Free Software Foundation to define software that has openly available source code and is freely modifiable. The term Open Source was introduced by members of the free software community who were concerned that 'free' in the English language is ambiguous and can mean both gratis and libre (Stallman countered that Free Software is free as in freedom, not as in free beer). The Ubuntu website states that the project supports both definitions.

2 This goal is stated in the Ubuntu bug tracker as Bug no. 1 entitled “Microsoft has a majority market share”– the bug “which Ubuntu is designed to fix”. The bug report has attracted several hundreds comments confirming and supporting that goal. Cf. <https://bugs.launchpad.net/distros/ubuntu/+bug/1>

from the Debian *Linux Distribution*³ upon which Ubuntu is based, they developed a system so technically promising and easily available (both on-line and through free world-wide shipping sponsored by Canonical) that it has spread beyond the circles of Free Software developers and into the IT mainstream.

It was always Shuttleworth's vision to build Ubuntu as a community rather than a company (his vision for Canonical is to sell support contracts on Ubuntu), and he asked the developers to continue working together on-line from their homes in an openly accessible, semi-egalitarian community with Shuttleworth as *Self-Appointed Benevolent Dictator For Life*⁴ at its centre, using their previous experiences in other Free Software projects as a base upon which to shape Ubuntu as a community. With the success of the Ubuntu system, the on-line community soon began to attract much interest from the thousands of developers whose software is integrated into Ubuntu. These projects are commonly referred to as the *Upstream* for Ubuntu, to indicate the flow of software technology or data from the original developers *Downstream* into Ubuntu.

Within 2 years of its initiation, the community has attracted more than 80 active developers, 20 of which are employed by Canonical, as well as volunteer translators, documentation writers, interface designers and grassroots advocates. These community members are predominantly male ethnic caucasians between 20 and 35, mostly, but not exclusively, university students or (often) university-educated IT-workers, mostly with comfortable middle-class backgrounds in the first world countries in Europe, North America and Oceania where they have had exposure to computing technology from an early age. They share a passion for technology and have the economic and social capital to access or buy technology to fuel this interest. In their everyday work on Ubuntu, they coordinate most of their shared efforts through the many means of communication afforded by the Internet, whereby their community is demarcated by technological infrastructure rather than by physical location. Depending on their specific interests, they meet at Free Software conferences around the world, but most of the community only convene at the bi-annual Ubuntu conferences sponsored by Canonical (cf. Lloyd 2006b, FLOSSPOLs 2006).

Though some Free Software hackers are very politically conscious and see clear political ends to their work, the Ubuntu project (and most other Free Software projects with it) is a politically agnostic entity,

3 See the glossary in Appendix A for definitions of the emic technical terms used within the Ubuntu community.

4 It is not unusual for founders of Free Software projects to assume “benevolent dictator” roles to provide a clear path for making decisions and settling disputes in an otherwise egalitarian community. After all, with Free Software, if people dislike your leadership enough, they will simply fork the project – that is: Take a complete copy of the source code and maintain it separately, away from the influence of the original project leaders.

allowing each contributor to define his own interest and motivation for contributing (cf. Coleman 2004). It is this passionate freedom of interest in and use of the technology that is the key to understanding the diverse motivations and passions that drive the Free Software movement, as noted by Coleman (2005) and Kelty (2002, 2004, 2005).

Aim of the fieldwork

The focus of my fieldwork was to study how the Ubuntu hackers' technical relationships with and social relationships through the computer define the social workings of the project. In my fieldwork proposal (Lloyd 2006a) I originally defined the aim of my study as

Firstly, to examine how the Ubuntu hackers use and relate to the computer in their work and everyday life as a means of intellectual pursuit; secondly to examine how they maintain social relations and coordinate their work and what part the computer, especially programming languages and technical jargon, play in this exchange.

This focus on the computer and programming languages as tools and means of communication was born from my hypothesis that hackers not only program the computer but also encode it with their cultural values, thus shaping ordinary users' experience of the computer. I hoped that by examining hackers' own computer use and technical exchanges I could find ways to bridge this cultural gap. But I soon found that the practice of programming with its technical jargon and programming languages stems from 50 years of computing tradition which shapes the practice of hackers as much as it indirectly shapes that of the ordinary computer users. To the hackers, the concepts contained within the technical jargon are precise designations of the abstractions with which software is built. The history and design of these tools do not attract active discussion within the community, much like the way anatomical denominations and specific medical tools are generally taken for granted within the medical profession..

Instead, I chose to limit my focus from the entire unbounded complexity of the networked computer to the nexus of the Ubuntu on-line community: The Ubuntu system as a technical artifact, and the technical infrastructures through which they coordinate and maintain the production of that artifact. By “technical artifact” I mean the layered complexity of software that is developed, maintained and produced by the Ubuntu on-line community – both in the form of a Installation CD and in the installed system itself through which hackers interact with the computer.

I define “technical infrastructure” as all of the technical means necessary for the development of Ubuntu, including communication channels such as *mailing lists* and collaborative development

designs and systems such as *software packages* and *Revision Control Systems*.

And I define the “Ubuntu on-line community” only as those users of the Ubuntu system who actively participate and contribute in the shaping and developing Ubuntu on-line – not only technically, but also socially through the enculturation of new contributors and adoption and negotiation of the formal community goals and social structures. This definition includes Canonical and its employees as active stakeholders at the centre of the Ubuntu community.

This change of focus led to a redefinition of the aim of my fieldwork:

Firstly, to examine how the Ubuntu on-line community uses and relates to the Ubuntu system as a means of intellectual pursuit and as a shared technical artifact; secondly to examine how they maintain social relations and coordinate their work through the technical infrastructure, especially with regards to how this infrastructure defines and is defined by the community.

This new focus required a reworking of my original research questions⁵:

0) How do the participants define the Ubuntu on-line community?

0a) For what reasons – intellectually, socially and in terms of proficiency – do hackers join the Ubuntu community?

0b) What goals, commitments and ideas are shared in the Ubuntu community?

1) In what ways do the Ubuntu hackers relate to the Ubuntu operating system itself - both individually and as a shared project?

1a) What criteria and elements do hackers value when programming and working with Ubuntu?

2) How do hackers coordinate their individual efforts in an Open Source software project such as Ubuntu?

2a) What part does specialized technical infrastructure such as Revision Control Systems and Software Packages play in defining the Ubuntu community?

2b) To what extent do hackers come to encode and share ethics, sociality and reciprocity through such technical infrastructure?

The practical course of the fieldwork

The distribution of the Ubuntu hackers necessitated a multi-sited fieldwork where I alternated between periods of participating on-line and periods travelling and visiting the hackers – both at Free Software

5 My original research questions as stated in my Fieldwork Proposal (Lloyd 2006a) were the following:

0) How do the participants define the Ubuntu on-line community?

0a) For what reasons – intellectually, socially and in terms of proficiency – do hackers join the Ubuntu community?

0b) What goals, commitments and ideas are shared in the Ubuntu community?

1) In what ways do the hackers relate to and perceive the computer itself?

1a) How do hackers use programming languages as a tool and as an element of interaction and reciprocity with the computer?

1b) What criteria and elements do hackers value in designing and programming a computer program?

2) How do hackers coordinate their individual efforts in an Open Source software project such as Ubuntu?

2a) What part do programming languages, specialised computer jargon and metaphors play in the social relations between hackers?

2b) How do hackers come to share ethics, sociality and reciprocity through the computer?

conferences where the on-line communities convene at hotels or universities for a week at a time, and in their homes where most hackers do their Ubuntu-related work, regardless of whether it is a pastime hobby or a full-time job in Canonical's distributed work model.

As noted in the chronology of my fieldwork (Appendix C), I participated solely on-line for the first two months following the announcement of my fieldwork (Appendix B), slowly getting accustomed to the many technical means of interaction and the succinct, humorous tone of the on-line banter. In that time I also developed an on-line survey to gather basic statistics of the Ubuntu community (cf. Appendix D, Lloyd 2006b) which I prepared to present at the Ubuntu Developer Summit in Paris in mid-June.

This conference felt like the beginning of the fieldwork proper, since it was my first opportunity to relate to the hackers in-person as individuals, something I had found difficult due to my lack of the technical aptitude necessary to participate in many of the exchanges on-line.

Following the Paris Summit, I sent out a fieldwork visit request (Appendix E) to 22 of the core Ubuntu developers I had met in Paris, roughly two thirds of whom were employed by Canonical. Over the summer I used the data I had gathered to refocus my fieldwork towards technical infrastructures, and between August and November, I spent gradually more and more time visiting the hackers in their own working environments, participating in their everyday life and interviewing them, as well as taking part in the interactions of the whole community on-line. I visited a total of 17 developers in 5 different countries in Europe and North America before ending my fieldwork at the following Ubuntu Developer Summit in Mountain View, California in early november 2006, which gave both coherency and closure to the project.

Fieldwork methods and results

Naturally, the methods I used and the data I gathered depended whether I was in an on-line or in-person context.

On-line methods

The defining trait of life on-line is that it is textual. Not only is all of the technical construction and maintainance of systems and infrastructure abstracted into text, but all of the on-line communication and socialization is also reduced to a textual dimension void of visual clues and rapport, allowing each individual complete control of how to present themselves through these limited means. Any distrust inherent in such circumstances is partly negated by the community's focus on actual contributions to Ubuntu and the formalized reciprocity of the GPL, rather than on-line rhetoric. Trust is built through shared collaborative work on-line, or by meeting and socialising with community members at

conferences.

The Ubuntu community uses its textuality as a strategy to further trust and attract new contributors by keeping its social interactions public: All emails sent to *mailing lists*, all the logs from all of the *IRC chat channels*, all the individual changes to each *wiki* page, every *blog* update on the *Planet Ubuntu* webpage, every issue recorded and discussed in the *bug tracker*, every *software package* upload and every *revision* of the source code is tracked, recorded and publically archived on-line. Though every community interaction of the volunteers and developer community is public, the dealings of Canonical which often shape the direction of the project is less so, naturally attracting mistrust both inside and outside of the community.

Because of the radical public nature of all of the textual interactions of the community, my presence as an observer and participant did not worry my informants. My on-line presence only mattered as much as the contributions I made to the community.

On-line participant observation

It is easy to passively observe – to read – such a textual community, but in order to participate I began contributing my own work and take part in shaping Ubuntu alongside other community members.

Because of my limited technical abilities I worked with the Documentation and Wiki teams, organizing and submitting *patches* – IT jargon for improvements – for the system documentation rather than the system itself, developing a feel for how the everyday exchanges and work in the Ubuntu on-line community is shared through the technical infrastructure (Q.2a).

Through my experiences contributing, I came to understand the discussions of which elements Ubuntu hackers value in the Ubuntu system, and how they use and relate to it individually (Q.1, Q.1a) and through that a tacit understanding of where conflicts tend to arise, and how charismatic leadership or huge amounts of development work can defuse them (Q.2).

I found that as I contributed to and used Ubuntu myself, my stake in Ubuntu grew as well as my feeling of empowerment in being able to take part. This made me focus on how the whole technical infrastructure by design seeks to promote the open sharing of ideas and knowledge, thus propagating the shared ethos of software freedom within the community (Q.2b).

Every day, as I read through the readily available textual data automatically on offer, I focused on cases with discussions on commonly recurring themes such as the ones mentioned above and in this way I slowly narrowed down the wealth of data to a number of clearly defined cases to use as a basis for analysis.

Documents

In many of the discussions, I noted how often documents such as the "Ubuntu Code of Conduct", the "Ubuntu Philosophy" and the "Ubuntu License Policy"⁶ are referred to as the agreed ideals of the community, and I carefully read over these documents to see how these ideals matched those of the members of the community in their everyday negotiations of those ideals (Q.0b).

Also, I examined the role and histories of the "Specification" documents which the community use as design frameworks for discussing the implementations of new features and elements of the Ubuntu system. Reading and contextualizing these documents through the elaborate Ubuntu archiving of revision histories and related discussions allows for a thorough exploration of the technical values the Ubuntu hackers value in the system (Q.1a), and I found how fluid the distinction between document and discussion can be in such a pragmatic and textually oriented community.

Quantitative Survey

I developed an on-line web survey in order to gather quantitative data on the basic statistics of the Ubuntu contributors as well as their computing habits, their use of the various means of on-line participation in the Ubuntu (Q.0) and their individual interests in Ubuntu - both as a technical artifact and as a community (Q.0a, Q.1).

I received around 300 responses, which, although confirming the trends of the recent FLOSSPOLS survey (FLOSSPOLS 2006) with regards to the gender, location and technical backgrounds of the community, most significantly described the fluid, pragmatically casual, upstream edges of the community rather than the solidly dedicated core I had found at its centre, underlining how the various channels of communication are used differently within the community, again directing my attention to the part the technical infrastructure plays in the community (Q.2a, Q.2b). The core developers I later interviewed had simply not noticed or reacted to the survey, and again when I published the survey results on-line (cf. Lloyd 2006b), they did not comment.

In-person methods

The individual, casual and often asynchronous on-line interaction was offset by the focused in-person interaction. I found, like Gabriella Coleman has, that on-line hacker communities constitute themselves through the prolonged in-person socialization at conferences hackers playfully call the "extended band-width" of meeting face-to-face. It is these events that enable the community to come together as such and create personal, empathic friendships to be augmented through on-line means (cf.

⁶ These documents can be found at <http://www.ubuntu.com/community/conduct>, <http://www.ubuntu.com/ubuntu/philosophy> and <http://www.ubuntu.com/ubuntu/licensing>, respectively.

Coleman 2005: 312ff).

In-person participant observation

The five – three Ubuntu and two upstream – conferences I went to allowed me to meet the developers on equal terms since all of the participants expect to meet people they have worked with, but can't recognize. To the hackers, the conferences are precious moments of intense in-person social interaction and conflict resolution, and I participated as community member, taking part in the discussions on technical and communal issues and observing the pragmatic manner in which the hackers discussed, proposed and implemented solutions (Q.1, Q.1a). My data reflects this in the form of meeting minutes, jotted observations and photos – more the notes of a community member than an anthropologist. Yet from these notes I have a sum of comparative data, showing the cultural and social interconnections between Ubuntu and its upstreams in the rest of the Free Software eco-system and how technical infrastructure often becomes the main boundaries between projects (Q.2, Q.2a).

Casual interviews

During the conferences I met and talked casually with a lot of different individuals. I introduced myself as an anthropologist studying the Ubuntu community and never once was I asked to explain what an anthropologist is, as most Free Software developers continually reflect on their role within the community structures and thus have begun to take the academic interest in their projects for granted. Many hackers, especially those with academic backgrounds, felt some kinship with me and sought eagerly to distance themselves temporarily from the community to offer their own observations and analysis of the community, as well being very interested to hear my observations. This discrepancy between this self-reflection and the apparent disinterest on-line which seems associated with social circumstances of the conference where hackers generally spend much time discussing and considering community issues, while the on-line fora generally are dedicated to technical discussion and coordination. This highlights the high level of education and academic interest within Free Software, which I anticipated and built on by sharing in these reflections while keeping a continuous anthropological focus which the hackers neither sought to or could maintain outside of the specific discussion.

Since these informal conversations, which I dubbed "casual interviews", often took place after-hours, and usually while drinking alcohol,⁷ I rarely took notes, but I did use them as an opportunity to inquire

⁷ This may raise some ethical concerns about the role of alcohol in interviews, but since all of the people I interviewed generally are very reflective upon their role in the community, I felt that the alcohol merely brought forth those views with a little more candour. In any case I was careful not to mention the details of those conversations to other informants.

about the informants' background with Free Software and about their opinions on key communal and ideological issues (Q.0, Q.0a, Q.0b). The narratives and observations they supplied was already scrutinized under their own analytical gaze before being passed on to me, showing how these hackers continually seek to master the dynamics of the communities of which they are part, as if it is a mathematical puzzle to be solved.

“Think-aloud” protocolling

I used "think-aloud" protocols (Lethbridge 2005) sitting with the individual developer and letting them run through one or two of the tasks on their to-do lists and let them explain step-by-step their concerns and considerations in solving those tasks. This set the individual development tasks into context within the greater whole of the continuous development process of Ubuntu and helped compensate for my lack of technical expertise, and while it proved invaluable for my understanding of the actual development work and the work processes and social interactions that they necessitated (Q.1, Q.1a), it also exposed just how much my understanding of these processes was mediated by the developers themselves, and I sought to alleviate this mediation by moving my focus from their specific tasks to their shared interactions (Q.2, Q.2a) which I was better suited to study.

In-depth interviews

When visiting the individual developers in their homes, most of my efforts were centred around the in-depth interview. I usually stayed with each developer for a few days, working alongside them and participating in their daily routines. The developers were generally very hospitable, yet I was very conscious about not letting my visit disturb their daily routines too much, since that would to an extent compromise the everyday nature of the setting I sought to study. Being in their homes and meeting their families and friends brought all of the loosely defined on-line and conference world into a well-defined real-life situation which I fleshed out through several hours of interviewing, delving into the informants' background, especially with regards to computers and Free Software (Q.0, Q.0a).

I then led a humanly mediated computer interview (Markussen 2002) using my extensive question guide (cf. Appendix F) to explore the digital space of file hierarchies, social contacts and links contained within the informant's computer, with the informant as a guide presenting his configurations and data (Q.1, Q.1a), and how he uses the technical infrastructure to share and coordinate his work with the other developers (Q.2, Q.2a). Since the informant's use of the computer was mediated through himself in this way, it is difficult to separate a hacker's actions from his descriptions of his actions, so I asked for examples of concrete situations and work cases where I could ask the informant to walk me through the on-line paper trail he had left behind in order to see how he related to his own interactions

on-line.

Hackers relate to their computers much like musicians relate to their instruments. It becomes an extension of themselves and the work they do, and though they tend to be very reflective about how they have configured it to best suit their needs, they also use it to shield their attention from distractions and towards the work they want to do. Visiting the hackers in-person removes this barrier, and not all my informants proved to be entirely comfortable with this unusual proximity and answered my questions as succinctly as they could.

Not only did the in-depth interviews, some of which I recorded in their entirety, allow me to gather a vast comparative array of data on the individual hackers' perception and physical use of the Ubuntu system, but they also allowed me to study the differences between how they compose themselves on-line compared to their presence in the home.

Analytical perspectives

I venture that it is necessary to examine Ubuntu both as a technical artifact, as an on-line community and as technical infrastructure in relation to one another to win insight into the social workings of the Ubuntu community. Based on this, my thesis will focus on the following layers of practice within the Ubuntu community:

1 - The Ubuntu operating system as a shared technical artifact

By using Bourdieu's ideas on the Kabyle houses as contested spaces (Bourdieu 1977) along with Ingold's ideas of *dwelling* and *building* (Ingold 1997, 2000) I hope to examine how the individual hacker adopts the Ubuntu system as a personal mental workspace that can be negotiated, decorated and extended. Further, I will also use Winograd & Flores' ideas of the computer as tool *ready to hand* (Winograd & Flores 1990) to examine the practices shared by the Ubuntu hackers through the common artifact – the Ubuntu system – and how the shared similar experiences, similar frame of mind that it affords shapes the community. I hypothesize that the individual and shared negotiations of the Ubuntu's default settings contains a key to understanding how the Ubuntu hackers relate their technical and ideological values to the Ubuntu system (Q.1, Q.1a).

2 - The formal social structures of the Ubuntu on-line community

I will seek to examine the formal social structures of the community as defined not only by the community members themselves, but also by the company funding it, and how these structures demarcate the otherwise openly fluid boundaries of the community. I will examine what role corporate and monetary interests play within the community compared to contemporary fluid communities, e.g.

the anti-globalization movement (Q.0).

I will also examine the formalized general reciprocity of the GNU GPL and the leadership roles within the project based on Marshall Sahlins' work on reciprocity and oceanic big men (Sahlins 1972) in relation to Leach and Kelty's recent work on multiple ownership and formalized reciprocity in F/OSS communities (Leach 2005, Kelty 2002, 2004, 2005). Particularly I will use my own experiences combined with my interview data to examine how new contributors in their enculturation and integration into these social structures negotiate between these idealistic values and their technical pragmatism (Q.0a, Q.0b).

3 - The practice and technical infrastructure of working together on-line

Most centrally, I will focus on the Ubuntu community as a *community of practice* (Wenger 1998) that not only shares common interests and ideals, but also the development of tools and knowledge. I will examine the explorative and pragmatic practice of Free Software development through Bruno Latour's ideas on Actor Network Theory (Latour 1987, 1993) to track the practice of software development on-line as technical and social information controlled through a technical infrastructure that is not only negotiated and used in these practices but also play its own part in shaping and defining the community – and thus the practices – itself (Q.2, Q.2a, Q.2b).

Though the data from my "think-aloud" protocols and my in-depth interviews has given me a good basis for understanding the practices of the community, defining and analyzing these will my biggest challenge, and more than anything this will require me to synthesize my on-line and in-person experiences into a coherent whole.

Appendix A – Glossary

This is a list of emic terms and concepts used within the on-line Free Software communities and in the Ubuntu community in particular.

Blog

Blog is short for *weblog*, a type of frequently updated website consisting of dated entries arranged in reverse chronological order so the most recent post appears first. Typically, weblogs are published by individuals and their style is personal and informal. Many Ubuntu developers have their own blogs, discussing a wide range of topics, but especially software development.

Bug Tracker

A system for receiving and filing bugs (programming errors, design flaws etc.) reported against a software project, and tracking those bugs until they are fixed. These reports can be commented by other developers, and the status of the bugs can be updated. Ubuntu uses a central Bug Tracking System called Malone.

Code of Conduct and Developer Guidelines

These are typical documents or social contracts within Free Software development. A hacker interested in working with the project has to verify his identity by digitally signing the Code of Conduct thus accepting these basic rules in order to participate in the project as proper members of the community.

Downstream

In IT jargon, Downstream indicates the receiver of a flow of software technology or data from an upstream source – usually the producer. Ubuntu is the downstream for Free Software projects such as GNOME and KDE, whose software the Ubuntu developers integrate and ship as part of the Ubuntu release.

Free Software

The term Free Software preceded the term Open Source Software, and was first used by the hacker Richard Stallman and his Free Software Foundation to define software that has openly available source code and is freely modifiable. The term Open Source was introduced by members of the free software community who were concerned that 'free' in the English language is ambiguous and can mean both gratis and libre (Stallman countered that Free Software is free as in freedom, not as in free beer). The Ubuntu website states that the project supports both definitions.

Free Software Foundation

The foundation founded by the early F/OSS hacker Richard Stallman to spread the concept of Free Software. See F/OSS.

GNU General Public License

The GNU General Public License is the legal license under which the Linux kernel and thousands of other F/OSS applications are licensed. It is the license created by Richard Stallman to embody the main ideas of Free Software. The GNU GPL is designed to guarantee all users and developers of the licensed program the following freedoms:

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this.

It is this license that first codified the general reciprocity on which the F/OSS is built (cf. Kelty 2002). See <http://www.gnu.org/philosophy/free-sw.html> for further details on Free Software).

IRC Channel

Internet Relay Chat is a protocol that allows for chat messages over the internet. An IRC channel is chat “room” of sorts, where, after joining the channel, your messages are broadcast to everyone listening to that channel. The Ubuntu project has 25 official development IRC channels (all in English) and 29 channels for various localized languages. These are realtime community discussion and meeting fora.

Kernel

The core of a computer operating system. It is the program that delegates resources to the other components and processes necessary for everyday use of the computer. Linux is probably the best-known F/OSS kernel, though other operating systems such as Apple's Mac OS X uses a F/OSS kernel called Mach.

Linux Distribution

A Linux distribution is a computer operating system comprising various Free Software components, such as the Linux kernel, the Open Office suite and the Mozilla Firefox browser – to name but a few of the most well-known. All of these have been developed in various Free Software projects, but are put together into a working operating system by the developers of the distribution. Ubuntu is based on an older Linux distribution called Debian, and still have strong ties to that development community.

Mailing List

A mailing list is a discussion group that occurs via mass e-mail distributions and to which individuals can subscribe to receive all the e-mails sent to that list. The Ubuntu project has 62 mailing lists, around half of which are localized language lists and the other half are development lists.

Newsgroup

An Internet discussion group that is available either through a newsreader program or through a web browser interface. There are more than 20 Ubuntu web fora at [<http://www.ubuntuforums.org>] with more than 60.000 registered users.

Patch

In computing, a patch is a software update meant to fix problems with a computer program. The term has since been extended not only to cover solutions to problems within programs, but to the fixing of all kinds of issues including replacing graphics, improving usability or performance, rewriting documentation or improving processes.

Planet

A Planet is a blog aggregation program used to combine the posts from several different blogs unto one single blog web page. This aggregation gives easy access to the latest posts of all of the different contributors, and many central Free Software projects such as KDE, GNOME, Debian and Ubuntu all have their own Planets where the blogs of the project developers are aggregated. Currently, the Planet Ubuntu [<http://planet.ubuntulinux.org/>] aggregates the blogs of 82 Ubuntu members. Me included.

Release Schedule

All computer programs are released with version numbering to make it easy to see how the software has changed from version to version. The Ubuntu project has a release schedule for the release of a new version of Ubuntu every six months, in April and October. Maintaining such a tight release schedule is usually not a given thing in Free Software production.

Revision Control System

Revision control (also known as version control) is the management of multiple revisions of the same unit of information, especially source code, to track changes made to this information. It also allows commenting and reverting

to previous versions of the code. The Ubuntu project uses a revision control system named Bazaar.

Software packages

A software package is a collection of source code prepared by a Free Software for easy installation with a relevant Linux Distribution. At present there are two main software package formats: **.rpm** and **.deb**. Ubuntu uses the **.deb** format which it has inherited from the Debian Linux distribution. The software package format allows for easy distribution of source code and divides the volunteer developers' work into manageable chunks since they can divide maintenance of the software packages between themselves.

Source code

Source code is the semi-legible boolean logic which the computer programmer writes in order to instruct the computer through a computer program. In order for this code to be executed on the computer, it needs to be translated into binary code. This is usually done by programs like a compiler or an interpreter.

Ubuntu Developer Summit

The Ubuntu Developer Summit is a biannual gathering of most of the core developers in the Ubuntu community. These events are organized and sponsored by Canonical, and alternate between American and European venues.

Upstream

In IT jargon, Upstream indicates the initiator, usually the producer, of a flow of software technology or data. Ubuntu is the upstream for a number of derivative distributions such as Mepis or Guadalinex that customize and specialize the general Ubuntu distribution to their specific needs.

Wiki

A wiki is a type of website that allows users to add and edit content, combined with a system that records each individual change that occurs over time, so that at any time, a page can be reverted to any of its previous states. The Ubuntu project use its wiki [<https://wiki.ubuntu.com>] extensively – including personal profiles, goal specifications, documentation, conference and meeting info.

Appendix B – Fieldwork initiation announcement

Email sent to the Ubuntu Sounder mailing list on 17:27 20/04/2006.

Hello all Ubunteros,

My name is Andreas Lloyd and I am a graduate student at the department of Anthropology at the University of Copenhagen, Denmark. Having used and enjoyed Ubuntu since november 2004, I have become very interested in the social workings of Free Software projects, and I wanted to combine my interest in F/OSS projects with my graduate studies. Therefore I propose to start an anthropological fieldwork study of the Ubuntu development community.

As I'm not much of a computer expert myself, I've been considering various other ways to contribute to the Ubuntu community. I've spent some time contributing to the Ubuntu documentation and the Danish translations, but I believe that it would be better for me to help improve the project by examining it from an anthropological perspective. You may have heard of American anthropologist Gabriella Coleman's work in the Debian community[1], and it is this sort of studies of how F/OSS projects are governed and maintained that I take as my inspiration.

With Ubuntu's "Linux for human beings" catchline, and its much-mentioned Bug #1 [2], the project seems to have a clear goal of developing a F/OSS operating system for a wider user base – especially in the Third World. With this goal in mind, I find it central to examine the way that Ubuntu developers perceive, use and talk of computers, as it is my hypothesis that the shared cultural and social values and ideas of the developers are shaping the way the average user perceives and uses the computer. Take, for instance, the fact that Ubuntu is a distribution of Linux whose basic shape and form is inspired by Unix – an operating system whose cultural heritage originates from an age when there were few computers and no end users - and continues to shape the way both users and developers perceive and use the computer.[3]

I am especially interested in how and to what degree social conceptions and jargon concerning computing technology govern the way we use it, and I hope that this fieldwork will help uncover new perspectives on how software developers encode the computer and the software they write with their own social and cultural values and ideas.

One of my key interests here is the interplay between developers and users in the community - especially with regards to the development and discussion on usability issues such as User Interface Design, Internationalization, Localization and Accessibility which seem to rarely receive much attention in F/OSS projects. By studying the way the developers work together and discuss these issues, I hope to pinpoint some of the problems that can arise between users and developers of Free Software. And furthermore, I hope that my fieldwork will help create more focus on a field of study which has received very little social scientific research attention so far.

A fieldwork study such as this one is a mandatory part of my graduate studies and will be the basis on which I will write my Master's Thesis. Initially, it was my plan to fit my fieldwork along a complete Ubuntu development cycle as the Ubuntu 6 month release cycle matches the average length of such a fieldwork quite well. I had planned to follow the now-codenamed Edgy Eft release cycle running from April to October 2006. But with the postponed release of Dapper Drake and the related shortening of the Edgy Eft release cycle, I am now ready to begin my fieldwork ahead of the new schedule. But in order to have the full 6 months in the field, I would like to begin the fieldwork soon after the date of Dapper's originally planned release – which is today!

This may seem like short notice, but in an online context it is rarely any good announcing a project until you're ready to follow through. Traditionally, anthropological fieldwork involves travelling to some remote part of the world, and spending a long period of time immersed in the local culture, learning their ways by taking part in their everyday life. But since the Ubuntu project is not centralized in any single location and has volunteers and developers spread all over the globe (though primarily Europe and North America), I will seek to do both

1. an online fieldwork and participation in the many digital fora and means of information exchange that used by the Ubuntu community: discussing on IRC channels and the mailing-lists, helping with bug

triage in Launchpad, reading blogs and writing documentation and suggestions in the Wiki.

2. an in-person fieldwork focusing on visits to individual developers where I will spend some time interviewing, observing and participating their daily life and work routines around the computer in order to examine how the development work takes place first-hand. Along with this, I will go to the developer's summits – such as the one announced to take place in June – and the few “sprints” in order to meet the developers and study how they meet each other to create and develop the personal and social ties which are the basis of the online collaboration.

This form of “multi-sited” fieldwork coordinated through the Internet has been developed by anthropologists in the last ten years, as it reflects the fragmented and globalized world which the discipline has as its object of study. I have received some grant funds to help finance these in-person field trips, so there it will not become any economical burden for the Ubuntu project.

Furthermore, as is usual practice with anthropological fieldwork data, all the material that I gather during the course of the fieldwork will be anonymized – unless the interviewed informants wish otherwise [4]. Also, I will make sure to present all of my findings to you, but please be fore-warned that an anthropological fieldwork takes time – and there are rarely any easy answers.

You are all most welcome to contact me (contact info below) – both those of you who may have questions regarding the fieldwork, or those who already now know that they do not want to take part.

If you are interested to know more about the theoretical basis for the fieldwork, I can send you the 10-page fieldwork proposal upon which the department of Anthropology at the University of Copenhagen has approved my fieldwork. It is rather full of anthropological jargon, but does explain the my project in greater detail.

If you are interested in knowing more about me and my academic background, feel free to read my weblog at <http://www.alligevel.blogspot.com/> or my Ubuntu wiki page at <https://wiki.ubuntu.com/AndreasLloyd>. I will also be participating in the Bug Day tomorrow, and will be online in the Ubuntu IRC channels under the name “lloydinho” - feel free to ask me questions there as well.

Best regards,

Andreas Lloyd

email: lloydinho@gmail.com
launchpad: <https://launchpad.net/people/lloydinho>
blog: <http://www.alligevel.blogspot.com/>
IRC: lloydinho on network irc.freenode.net

[1] http://papers.ssrn.com/sol3/papers.cfm?abstract_id=805287

[2] <https://launchpad.net/distros/ubuntu/+bug/1>

[3] <http://www.joelonsoftware.com/articles/Biculturalism.html>

[4] <http://ethics.iit.edu/codes/coe/aanta-1998.html>

Appendix C – Course of fieldwork – April to November 2006

<p>Stage 1 April-May Weeks 16-20</p>	<p>Entering the field</p> <ul style="list-style-type: none"> – The spring release of Ubuntu, codenamed Dapper Drake and scheduled for release on the 20th of April, is delayed until the 1st of June as it needs extra work on language support and artwork polish. – In spite of this delay, I announce my fieldwork project to the community (Appendix B), and begin active and official fieldwork within the community with participation in on-line meetings on IRC, reading bug reports and version updates (this continues throughout the fieldwork). – I spend time getting accustomed to the interactions on-line and begin contributing work to the Ubuntu documentation Team. – Detailed investigation of Ubuntu documents and rhetoric.
<p>Stage 2 May-June Weeks 21-24</p>	<p>Quantitative Survey</p> <ul style="list-style-type: none"> – Developing and sending out a quantitative on-line survey to the Ubuntu community members to gain an overview of the basic statistics of the community and demarcate the field. – On the 1st of June, Ubuntu 6.06 aka Dapper Drake is released. My fieldwork begins proper with the initiation of the new, shorter release cycle for the the autumn release of Ubuntu codenamed Edgy Eft. – Gathering and analysing the data from the on-line survey and prepare it to be presented at the Ubuntu Developer Summit in Paris where the coming Ubuntu release will be planned.
<p>Stage 3 June Weeks 25-26</p>	<p>Fieldwork at conferences</p> <ul style="list-style-type: none"> – Participating at the week-long Ubuntu conference near Paris and meet the developers in-person. – Participating in the technical and communal discussions of specifications relevant to the new Ubuntu release. – Conducting casual interviews to better gauge how the developers themselves consider the community they are part of. – Participating in the week-long GUADEC conference near Barcelona, where many of the “upstream” developers for Ubuntu gather and discuss the future of the F/OSS desktop project called GNOME. – Participating in the talks and sessions at the conference, observing the differences between the different projects. – Conducting a few narrative interviews with a few of the GNOME developers to find out their views on and relationships with Ubuntu.
<p>Stage 4 July- August Weeks 27-28, 30-33</p>	<p>Continued online fieldwork and analysis</p> <ul style="list-style-type: none"> – Continued online fieldwork, further work with the Ubuntu Documentation team and other teams. – Organising and transcribing fieldnotes. – Refocusing of the fieldwork towards technical infrastructures and their shaping of the community. – Week 29 is vacation time. – Writing a “Contribute to Ubuntu” document for inclusion in the Ubuntu documentation.
<p>Stage 5 August-October</p>	<p>In-person fieldwork in Europe</p> <ul style="list-style-type: none"> – Participation in the week-long Ubuntu Developers' Sprint in Wiesbaden. Further in-

<p>Weeks 34-44</p>	<p>person participant observation and “Think-aloud”-protocolling.</p> <ul style="list-style-type: none"> - Two weeks spent visiting 7 Ubuntu developers located in Germany with in-depth interviews focused on the informants' computer (humanly mediated computer interviews) as well as “talk-aloud” protocolling. - Becoming an official Ubuntu member at the on-line Community Council meeting on Tuesday the 22nd of August. - Organising and transcribing fieldnotes - Participation in week-long “upstream” aKademy conference in Dublin. - 2 weeks spent visiting 6 Ubuntu developers in Great Britain using the same methodology. - A week spent in Oslo interviewing 2 Ubuntu developers using the same methodology. - On the 26th of October, Ubuntu 6.10 aka Edgy Eft is released.
<p>Stage 6 November Weeks 45-46</p>	<p>In-person fieldwork in North America</p> <ul style="list-style-type: none"> - Participation in Ubuntu Developer Summit at Google's headquarters in Mountain View, CA. - Participating in the technical and communal discussions of specifications relevant to the new release. Informal interviews regarding the development of the community and the project as a whole. - Visiting local Ubuntu developers in North America.
<p>Stage 7 November Week 47</p>	<p>Conclusion of Fieldwork</p> <ul style="list-style-type: none"> - Conclusion of fieldwork upon return to Copenhagen.

Appendix D – Ubuntu Census Survey announcement

Email sent to the Ubuntu Sounder, Ubuntu-Devel and Ubuntu-Users mailing lists on 12:37 16/05/2006. Also posted on the Ubuntu Web Forums *Ubuntu Café* Forum.

To all users and developers of Ubuntu,

To begin my announced anthropological fieldwork in the Ubuntu community [1], I have constructed a questionnaire to gather basic and relatively valid statistics on the community as whole. I call it a census survey since there's no official register of people contributing to Ubuntu [2]. In spite of the name, the survey will not be used to keep track of individuals and their contributions to Ubuntu. Instead, the objective is to examine statistically the membership of the Ubuntu community:

- Who are joining the Ubuntu community and for what reasons.
- How they use computers in general.
- How is their relationship to the F/OSS world in general.

I hope that as many members of the Ubuntu community as possible will find the time to fill out the questionnaire - this data will help give an overview of where the Ubuntu community is now, and hopefully make it clearer where it is going.

The questionnaire consists of 5 sections and a total of 65 questions (excluding the comment fields). It should take around 15 minutes to complete, depending on your level of involvement in the F/OSS world, and the number of comments that you find you want to add while filling out the questionnaire.

You can find the Survey Questionnaire here:

<http://www.surveymonkey.com/s.asp?u=855442126391>

All of the survey data I receive will be anonymized. Your name will only be used to file the questionnaire. Your privacy will be protected as according to usual anthropological practice [3].

I plan to have this data collected and analyzed to present the results to you at the Ubuntu Developers' Summit in Paris in June, and I will of course make the relevant results available on-line afterwards.

Feel free to send me any questions or comments that you might have regarding this survey or my fieldwork in general.

Thank you very much for your help!

Andreas Lloyd

-

<https://launchpad.net/people/lloydinho>

PS: Sorry for cross-posting - this is a necessary evil to reach as much of the Ubuntu community as possible, as the -announce mailing-lists are reserved for announcements critical to the entire community.

[1]: <https://lists.ubuntu.com/archives/sounder/2006-April/006113.html>

[2]: At the moment of writing, there are 1270 Launchpad members who have signed the Code of Conduct, thus qualifying for the Ubuntero badge. But there is most certainly many other Ubuntu users and participants who haven't signed the Code of Conduct, or who simply don't use Launchpad.

[3]: <http://ethics.iit.edu/codes/coe/aanta-1998.html>

Appendix E – Fieldwork visit request

Email sent to more than 15 individual core Ubuntu developers on 25/07/2006.

Hi [developer name],

It was great meeting you and the rest of the Ubuntu developers at the Summit in Paris in June. I didn't really feel that my anthropological fieldwork [<https://wiki.ubuntu.com/UbuntuFieldwork>] really started until I got to meet all of you in person.

As I said at the summit, my fieldwork consists of two separate elements: an on-line part and an in-person part. Until now, I have mostly been doing the on-line part, reading and contributing to mailing-lists, specifications and bug reports, getting a feel of the jargon and the way social interactions work through IRC, the Wiki and in the Web Forums.

But now, I would like to do begin the in-person part of the fieldwork properly, and that will require me to come and visit as many as Ubuntu developers as possible to observe and talk with you about how your work routines with the computer and how coordinate your efforts with the other developers and the relevant upstreams.

I have been working to tighten the focus of my fieldwork on how software developers use the computer - both as a tool and means of social communication, and how that use is shaping the way that they perceive use of computers in general - including those end-users for whom they design, and I believe that by interviewing you and observing your individual work routines, I will get a better insight in how considerations around Ubuntu development are discussed and solved.

You may well wonder what such a fieldwork visit would entail. My initial idea was this:

I come to visit you for a couple of working days. Normally, this wouldn't be very intrusive, but since most of you work at home, I do not yet know how you would prefer to do this. A visit would consist of several parts:

- taking a tour of your work space with whatever technology, gadgets and computers you may have gathered.
- interviewing you on relevant background information on your involvement with F/OSS and Ubuntu.
- going on a virtual tour of the computer with you as the guide, explaining the way that you have organized your data.
- sitting by you as you work as usual while you describe your actions at the computer in detail.

This may sound very formal, but that is not the intention. I expect it to be very relaxed and a lot of fun. I have tried this set-up with a few local Computer Science students I know here in Copenhagen, and they enjoyed not only getting to tell about their system, but also being surprised at some of their routines which they hadn't actively considered before. Hopefully the entire fieldwork, and the Master's Thesis that I will write based on it will gather all of these insights and yield some new ideas on how F/OSS developers work together and produce software for human beings.

The visit will take some time, most likely a couple of days, so that will be two days where you won't be working as effectively as you otherwise would. Therefore I want ask you in good time about when (and if!) it would suit you for me to come and visit.

I will be going to the Ubuntu Distro Sprint in Wiesbaden in August, and hope to spend some time visiting the German and French Ubuntu developers around then. Afterwards, I hope to go on a Fieldwork tour of the UK in early September - How does that sound to you?

Please let me know if you have any questions or concerns about the fieldwork or the concept of the visits.

Best regards,

Appendix F – Interview Question Guide

Firstly, to examine how the Ubuntu hackers use and relate to the computer in their work and everyday life as a means of intellectual pursuit; secondly to examine how they maintain social relations and coordinate their work and what part the computer play in this exchange.

Make them tell stories!

Gender

Age

Nationality

Location (current home)

Private Status and children

Level of education inc. eventual majors

Current profession

Personal background with computers

- **When** and **how** did you begin using computers? Experiences?
- What attracted you about computers initially?
- At what age did you begin programming computers comfortably?
- What educational background do you have with computers? Tradition, norms?
- Friends, mentors, fellow travellers?

Finding F/OSS

- **How** did you discover F/OSS? How did it feel to find such a community?
- What attracted you to F/OSS?
- Which projects have you contributed to - and for what reasons – intellectually, socially and in terms of proficiency?
- How has your own background (cultural, language-wise etc.) defined your participation in F/OSS?
- When and how did you join the Ubuntu community?
- How are you involved in Ubuntu?
- How much time do you spend working on (contributing to) Ubuntu in an average week?

Life apart from computers

- How do you balance your computing life with your everyday life?
- How do your family and friends relate to your job / hobby?
- How do you explain what you do to non-technical people?

Using and programming the computer as tool

- Taking a **grand tour** of the informant's computing life: Let the informant introduce his computer(s). The relationship between them, which one is his primary computer. Then focus on that one. The computer as a house – decorated and inhabited.
How many computers do you own, use or have access to? What functions do they fulfil?
- How much time do you use on computers (working, playing games, socialising online etc.) in an average week?
- Which operating systems do you use? Why?

- Unix/Linux vs. other operating system architectures?
- “Describe **how you usually use your computer**. Which programs do you use, how you've arranged them.” “**Think-aloud**” protocol – show the work you do with Ubuntu: Packaging, patching, bug reporting etc.
- Personal desktop setup and settings
 - How did you come about this setup? Which other variations have you tried or considered?
 - GNOME vs. KDE?
 - the command line
 - text editing programming (languages, design...)
 - configurations, scripts
 - structure – “how do you organize your files?”
 - Bazaar and revision control systems – branches, repositories?
- What are the relations between the various computers you use? Build machines, communication machines, servers, backups, test machines?
- When are you productive?
- Being in the zone? The feeling of flow?
- What are your goals with the computer? What do you hope to achieve with or through the computer?

Software and values

- What criteria and elements do you value when designing and programming a computer program? who do you imagine will read the code you write? Who is the audience?
- What do you like most about programming? What do you get out of it? What is the creative element? Is it an art? Examples of beautiful code?
- Responses from people who have used your code?
- Changelogs? Revision histories?
- who do you work with, how do you coordinate their individual efforts?
- Timezones? Cultural differences?
- ownership of the code? How do you imagine your code will be used?

On others' use of computers

Design philosophy?

- The Unix philosophy?
- helping the end user (who is that?) Do you help people with computers?
- imagining use cases
- Developer Guidelines (GNOME HIG, Debian Developers Guidelines etc.)
- discussing with other developers
- responses, bugs, wishes from the users
- Matthew Garrett:

“Given a choice between making it easier to configure something and making it unnecessary to configure it, we should always choose the latter. Having a lot of options that should "just work" makes it harder for people to find the (fairly small) number of options that *are*/ absolutely required.” [Sunder 24/05/06 11:22]

- What does this mean, exactly? And do you agree? Why?
- Sane defaults

The computer as social tool. Communication.

- email (mailing-lists, contacts, signed PGP keys...)

- IRC (channels, nicks, commands, notifies...)
- Who do you communicate regularly with over the computer? And how?
- Web Forums
- Blogs
- Wikis
- Bug trackers (reading, writing, commenting on bug reports, fixing them...)
- Revision Control Systems (reading other people's code, commit access...)
- “how do you coordinate your work with others?”
- jargon – metaphors, technical terms, hacker slang (differences, interests problems?)

Note specifications, discussions, bug reports that they have been working on. Ask about their **rationale**.

The Ubuntu Community

- Describe the Ubuntu community in relation to your role in it.
- The structure of the community. Draw a flow diagram, structure of the community. What defines it?
- What goals, commitments and ideas are shared in the Ubuntu community?
- Why do people join the Ubuntu community?
- Relationship with Upstream: General development upstream and Debian.
- The role of Canonical?
- Code of conduct? Governance? SABDFL?
- Teams? Leaders?
- Technical infrastructure?
- Release cycles, specs?
- Launchpad?
- Karma?
- Bazaar? Package management?
- What do you dislike about Free and Open Source Software?
- What do you dislike about Ubuntu?

The Future

- what challenges do you see in the future for Ubuntu?
- Where do you see F/OSS in 5 years? In the long term?
- How do you think computers will develop? What will they (continue to) affect society? Singularity?

Questions and Comments

References cited

- Pierre Bourdieu: *Outline of a Theory of Practice* (Cambridge: Cambridge University Press, 1977).
- Gabriella Coleman: "The Political Agnosticism of Free and Open Source Software and the Inadvertent Politics of Contrast". *Anthropological Quarterly* 77(3) Summer 2004.
- Gabriella Coleman: *The social construction of Freedom in Free and Open Source Software: Hackers, Ethics and the Liberal Tradition*. PhD. Dissertation, University of Chicago 2005.
- FLOSSPOLs (Free/Libre/Open Source Software: Policy Support) Project at the Maastricht Economic and social Research and training centre on Innovation and Technology: Reports on "Sustaining a software ecosystem with FLOSS: skills and local economic growth" and "Gender: Integrated Report Findings" (2006). Accessed at <http://flosspols.org/dissemination.php> on December 5th, 2006.
- Tim Ingold: *The perception of the environment: essays on livelihood, dwelling and skill*. London: Routledge, (2000) xvi, 454 pp.
- Tim Ingold: "Eight themes in the anthropology of technology". *Social Analysis* (special issue 1997, Technology as skilled practice, ed. P. Harvey) 4(1): pp. 106-138.
- The Jargon File (v.4.4.7 – updated October 2003) Accessed at <http://www.catb.org/~esr/jargon/> on December 7th, 2005.
- Christopher Kelty: "Culture's Open Sources" and "Punt to Culture" in *Anthropological Quarterly* 77(3) Summer 2004.
- Christopher Kelty: "Geeks, Recursive Publics, and Social Imaginaries" in *Cultural Anthropology* 20.2 Summer 2005
- Christopher Kelty: "Hau To Do Things With Words" (2002) – Paper accepted for publication by the JAI Press quarterly series *Knowledge and Society*, but omitted from the final publication. Accessed at <http://www.kelty.org/or/papers/Kelty.Hautodothings.2002.rtf> on January 5th, 2006.
- Bruno Latour: *We Have Never Been Modern* (Cambridge MA: Harvard University Press, 1993).
- Bruno Latour: *Science in Action* (Cambridge MA: Harvard University Press, 1987).
- James Leach: "Modes of Creativity and the Register of Ownership" in: *CODE – Collaborative Ownership and the Digital Economy* (Cambridge, MA: MIT Press, 2005)
- Andreas Lloyd: *The Ubuntu hackers' perception and use of computers*. Fieldwork proposal submitted at University of Copenhagen, 2006a.
- Andreas Lloyd: "Results of the Ubuntu Census Survey" (2006b). Accessed at <http://www.eskar.dk/andreas/output/Survey-intro.html> on December 5th, 2006.
- Timothy Lethbridge, Susan Elliott Sim & Janice Singer: "Studying Software Engineers: Data Collection Techniques for Software Field Studies" in *Empirical Software Engineering* Vol. 10, Issue 3 (July 2005) pp. 311 – 341.
- Ole Møller Markussen: "The Humanly Mediated Computer Interview." in *Den Vilde Tanke* No. 24 (October 2002), pp. 2-4.
- Marshall Sahlins: *Stone Age Economics* (Chicago: Aldine, 1972).
- Etienne Wenger: *Communities of Practice: Learning, Meaning, and Identity* (Cambridge: Cambridge University Press, 1998)
- Terry Winograd & Fernando Flores: *Understanding Computers and Cognition* (Norwood, NJ: Ablex Publishing, 1990 [1986]),